**Лекция №2. Основные языки, инструменты и библиотеки МО.**

1. **Языки программирования**

Сегодня существует большое количество программных инструментов для создания моделей машинного обучения. Первые такие инструменты формировались в среде ученых и статистиков, где популярны языки R и Python, исторически сложились экосистемы для обработки, анализа и визуализации данных именно на этих языках, хотя определенные библиотеки машинного обучения есть и для Java, Lua, С++.

Общая особенность всех языков, применяемых для машинного обучения – использование векторизации вычислений. Компиляторы (интерпретаторы) реализуют высокую производительность для векторизованных вычислений.

Алгоритмы работают не с отдельными ячейками данных, а с многомерными массивами, что увеличивает их производительность.

В некоторых языках векторизация встроена непосредственно в язык, в некоторых реализована с помощью библиотек. В частности, в Python векторизация реализована с помощью библиотеки NumPy.

**MATLAB**

Matrix Laboratory – пакет прикладных программ для решения задач технических вычислений и одноименный язык программирования, используемый в этом пакете. Пакет используют более миллиона инженерных и научных работников, он работает на большинстве современных операционных систем, включая Windows, Linux, Mac OS. Является пакетом №1 для анализа данных и машинного обучения.

DSL (Domain-specific language – язык, специализирующийся на определенной прикладной области) – язык пакета MATLAB ориентирован на математиков. Не предназначен для разработки полнофункциональных программных систем.

Пакет является проприетарным и платным. Существует свободно-распространяемый аналог – GNU Octave, язык которого в целом совместим с MATLAB, но который содержит меньше библиотек и отличается менее высокой производительностью.

**R**

R – язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом. Изначально был ориентирован на задачи математической статистики, но в настоящее время содержит большое количество пакетов для анализа данных и машинного обучения. DSL-язык, ориентированный на математиков. Не предназначен для разработки полнофункциональных программных систем.

**Python**

Высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Одним из основных преимуществ использования Python является возможность напрямую работать с программным кодом с помощью терминала или других инструментов типа Jupyter Notebook, который мы рассмотрим ниже. Машинное обучение и анализ данных – это в основном итерационные процессы, в которых данные задают ход анализа. Поэтому крайне важно для этих процессов иметь инструменты, которые позволяют оперативно и легко работать.

Еще одно преимущество Python состоит в том, что может использоваться и как язык для обработки данных и как язык для разработки приложений (веб-приложений). Это очень облегчает встраивание ML-решений в веб-приложения.

Написанные под Python библиотеки машинного обучения – самые популярные на данный момент.

1. **Инструменты и библиотеки**

**Jupyter Notebook**

Jupyter Notebook представляет собой интерактивную среду для запуска программного кода в браузере. В мире Data Science Jupyter-ноутбук уже несколько лет считается одним из популярных инструментов для анализа данных и быстрого прототипирования. Несмотря на то, что Jupyter Notebook поддерживает множество языков программирования, нам нужна лишь поддержка Python. Один из плюсов этого инструмента в том, что код можно разделить на кусочки и работать над ними в любом порядке. Например, написать скрипт и сразу посмотреть, как он работает. Остальные фрагменты кода при этом запускать не нужно, результат появляется тут же, под частью кода. В нём удобно экспериментировать с новыми идеями в режиме «зашёл-сделал-вышел», писать документацию и создавать аналитические отчёты.

**Google Colab**

Еще один удобный инструмент, который вам может понадобиться, если вы работаете с Python, – это Colab. Colaboratory, или просто Colab, позволяет писать и запускать Python в браузере. Он не требует настройки, дает вам доступ к мощности графического процессора, а результатами легко поделиться.

Одно из главных преимуществ Google Colab – возможность использовать GPU (графический процессор) или TPU (процессор для машинного обучения Tensor Processing Unit) для выполнения вычислительно интенсивных задач, таких как обучение глубоких нейронных сетей. Это делает Colab особенно полезным для работы с машинным обучением и искусственным интеллектом.

**PyCharm**

PyCharm – это мощная интегрированная среда разработки (IDE) для Python, которая может значительно помочь в реализации и развертывании алгоритмов машинного обучения. Она предоставляет удобный интерфейс и множество инструментов, которые могут упростить процесс разработки. Среди питонистов эта IDE вторая по популярности после редактора кода Visual Studio Code: как основную её используют 31% разработчиков.

**Аnaconda**

Дистрибутив Python, предназначенный для крупномасштабной обработки данных, прогнозной аналитики и научных вычислений. Anaconda уже включает NumPy, SciPy, matplotlib, pandas, IPython, Jupyter Notebook и scikit-learn. Есть версии для Mac OS, Windows и Linux. Это очень удобное решение и это тот дистрибутив, который мы рекомендуем пользователям, у которых еще не установлены пакеты Python для научных вычислений. Кроме того, сейчас Anaconda включает в себя коммерческую библиотеку Intel MKL, которой можно пользоваться бесплатно. Использование MKL (это происходит автоматически при установке Anaconda) может дать значительный прирост скорости при выполнении различных алгоритмов в scikitlearn.

**NumPy: готовые вычислительные алгоритмы и линейная алгебра для машинного обучения**

Данные в машинном обучении представлены числовыми массивами. Даже если мы работаем с картинками или естественной речью, они должны быть преобразованы в числовые массивы. В NumPy уже реализовано всё необходимое для этого: преобразование Фурье, генерация случайных чисел, перемножение матриц и другие сложные операции.

**Pandas**

Pandas – библиотека Python для обработки и анализа данных. Она построена на основе структуры данных, называемой DataFrame. DataFrame библиотеки pandas представляет собой таблицу, похожую на электронную таблицу Excel. Библиотека pandas предлагает большой спектр методов по работе с этой таблицей, в частности, она позволяет выполнять SQL-подобные запросы и присоединения таблиц. В отличие от NumPy, который требует, чтобы все записи в массиве были одного и того же типа, в pandas каждый столбец может иметь отдельный тип (например, целые числа, даты, числа с плавающей точкой и строки). Еще одним преимуществом библиотеки pandas является ее способность работать с различными форматами файлов и баз данных, например, с файлами SQL, Excel и CSV.

**Matplotlib**

Matplotlib – это основная библиотека для построения научных графиков в Python. Она включает функции для создания высококачественных визуализаций типа линейных диаграмм, гистограмм, диаграмм разброса и т.д.

**NLTK: разбираем естественный язык на части**

Один из ведущих инструментов для обработки естественного языка. По аналогии с тем, как NumPy упрощает линейную алгебру, NLTK упрощает парсинг текста, анализ тональности, структуры предложений и всё, что с этим связано.

**Scikit-learn**

Библиотека с более чем десятилетней историей содержит реализации практически всех возможных преобразований, и нередко ее одной хватает для полной реализации модели. Как правило, при программировании практически любой модели на языке Python какие-то преобразования с использованием данной библиотеки всегда присутствуют.

Scikit-learn содержит методы разбиения датасета на тестовый и обучающий, вычисление основных метрик над наборами данных, проведение кросс-валидации. В библиотеке также есть основные алгоритмы машинного обучения: линейной регрессии, основных методов кластеризации.

Эта библиотека имеет открытый исходный код и может использоваться как в исследовательских, так и в коммерческих целях. Spotify, например, сделали свою рекомендательную систему как раз с помощью scikit-learn. Работает в связке с SciPy, NumPy и Matplotlib. Все базовые функции типа кластеризации, классификации и регрессии, разумеется, на месте.

Дальше речь пойдёт о глубоком машинном обучении нейронных сетей. Здесь для принятия более сложного решения алгоритм учитывает различные факторы, пропуская входящие данные через множество слоёв нейронов.

Само собой, для этого нужно больше вычислительной мощности и данных для обучения. Например, для GPT-3 OpenAI насобирали датасет из 45 ТБ текстовых данных и отфильтровали его до 570 ГБ. Обучение модели стоило им миллионы долларов. При этом использовали они даже не весь текст.

**PyTorch**

Фреймворк [машинного обучения](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для [языка Python](https://ru.wikipedia.org/wiki/Python) с открытым исходным кодом, созданный на базе [Torch](https://ru.wikipedia.org/wiki/Torch). Используется для решения различных задач: [компьютерное зрение](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5_%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D0%B5), [обработка естественного языка](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B5%D1%81%D1%82%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0). Разрабатывается преимущественно группой искусственного интеллекта [Facebook](https://ru.wikipedia.org/wiki/Facebook).

**TensorFlow**

Это библиотека для машинного обучения, группы технологий, которая позволяет обучать искусственный интеллект решению разных задач. Заточена для решения задач бизнеса: пропускать через себя огромные массивы данных при хорошей производительности и с возможностью использовать модели на мобильных устройствах.

**Keras**

Keras – более высокоуровневый интерфейс для TensorFlow, который представляет собой библиотеку глубокого обучения для Python. Простыми словами, он создан, чтобы стать языком «С++ машинного обучения» для низкоуровневых фреймворков. Новичок может не думать, как реализовать тензорную алгебру, построить модель и прочее. Он просто воспользуется готовыми строительными блоками.